

MEMORANDUM  
RM-4540-PR  
MARCH 1966

AD629606

## THE CATALOG INPUT/OUTPUT SYSTEM

Martin Kay, Frederick Valadez and Theodore Ziel

CLEARINGHOUSE FOR FEDERAL SCIENTIFIC AND TECHNICAL INFORMATION			
Hardcopy	Microfiche		
\$3.00	\$0.75	71 PP	as
ARCHIVE COPY			

Code 1

PREPARED FOR:

UNITED STATES AIR FORCE PROJECT RAND

20050310002

The RAND Corporation  
SANTA MONICA • CALIFORNIA

MEMORANDUM

DM-4540-PR

MARCH 1966

THE CATALOG INPUT/OUTPUT SYSTEM

Martin Kay, Frederick Valadez and Theodore Ziehe

This research is sponsored by the United States Air Force under Project RAND—Contract No. AF 19(638)-1700—monitored by the Directorate of Operational Requirements and Development Plans, Deputy Chief of Staff, Research and Development, Hq USAF. Views or conclusions contained in this Memorandum should not be interpreted as presenting the official opinion or policy of the United States Air Force.

DISTRIBUTION STATEMENT

Distribution of this document is unlimited.

The RAND Corporation

1700 MAIN ST. • SANTA MONICA • CALIFORNIA • 90406

-11-

PREFACE

This Memorandum completely defines the format used to store data on magnetic tape according to a general yet flexible scheme described previously.\* It also describes the set of routines, called the Catalog Input/Output System, that has been implemented for writing data in the form of a catalog and for reading from a catalog into a high-speed core.

Computational linguistics has created a need for data files with characteristics significantly different from those required in other computer applications, and since several linguistic research groups have committed themselves to cooperative interchanges of data, this scheme should help meet the need. Several million words of Russian text and a Russian physics glossary have already been written in this format at The RAND Corporation. In addition, the Linguistics Research Center at the University of Texas, the Computation and Data Processing Center at the University of Pittsburgh, the Centre d'Etudes pour la Traduction Automatique at the University of Grenoble, and the RAND Linguistic Research Project all have plans to write additional data files in this format in the near future.

---

\*Martin Kay and Theodore Ziehe, The Catalog: A Flexible Data Structure for Magnetic Tape, The RAND Corporation, RM-4645-PR, October 1965.

-iv-

The routines of the Catalog Input/Output System, written in the MAP language for the IBM 7044, are currently in use both at RAND and in Grenoble. Extensions to the system for sorting and merging are being designed and implemented in RAND.

-v-

### SUMMARY

This Memorandum completely defines the format used for catalogs on magnetic tape and describes in detail the routines of the Catalog Input/Output System. Catalog maps, catalog data, and tape labels are written as logical records in a specially designed blocking format. Beginnings and ends of blocks, physical tapes, and catalogs are explicitly marked. The Catalog Input/Output System offers a variety of input/output unit-control operations in addition to the commands for reading and writing data in catalog format. The reading and writing commands are implemented on three levels: Level III handles individual data; Level II reads and writes logical records; and Level I processes blocks of information.

CONTENTS

PREFACE .....	iii
SUMMARY .....	v
Section	
1. INTRODUCTION .....	1
2. CATALOGS ON MAGNETIC TAPE .....	3
2.1. A Blocking Format .....	5
2.2. Catalogs and the Blocking Format .....	7
3. THE CATALOG INPUT/OUTPUT SYSTEM .....	17
3.1. Control Operations .....	20
3.2. Level III Operations .....	32
3.3. Level II Operations .....	43
3.4. Level I Operations .....	49
3.5. Logical Unit Table .....	52
Appendix	
A. CO-OCCURRENCE OF BEGIN AND END MARKERS .....	55
B. EXAMPLES OF BEGINNINGS AND ENDS OF CATALOGS AND TAPES .....	57

## THE CATALOG INPUT/OUTPUT SYSTEM

### 1. INTRODUCTION

Computer applications in linguistics, library science, and social science are creating a need for very large, intricately structured, and in some cases tentatively organized files of data. The catalog--a generalized format for data structures--is designed to meet that need. The plan behind this development and an introduction to the use of catalogs are given in The Catalog: A Flexible Data Structure For Magnetic Tape.\*

Computer programs are being designed and implemented to facilitate the use of catalogs. (This work is being performed jointly by the Centre d'Etudes pour la Traduction Automatique at the University of Grenoble and the Linguistic Research Project at The RAND Corporation.) The computer programs will:

- a) Facilitate partitioning, rearranging, and converting data from any source in preparation for writing the catalog.
- b) Format and convert data for printing on one of a variety of printers.
- c) Sort the data elements within a catalog and merge data from two or more separate catalogs.

---

\*M. J. Kay and T. W. Ziehe, The RAND Corporation, RM-4645-PR, October 1965.



- d) Restructure a file by rearranging the order of classes of data--catalog transformations.
- e) Address nodes in the structure, retrieve data from the structure, place new data into the structure, and add to or delete from the structure--file maintenance.

In the long term, provisions will be made for storing catalogs in and using them from any kind of storage device--disks and other random-access stores as well as serial-access magnetic tapes. Present programs are, however, being designed to use catalogs from magnetic tape. This form of storage is relatively inexpensive, essentially unlimited in capacity, widely used, and easy to transport between cooperating groups.

This document describes the basis on which the present development effort rests. Section 2 describes in detail the format adopted for catalogs stored on magnetic tape. Catalogs are written on six-channel (plus parity) magnetic tape at 800 bits per inch (bpi) in the IBM binary mode. Other densities can be used, 556 or 200 bpi, but a density of 800 bpi is assumed unless otherwise specified. Catalog tapes are written so that they can be read by the IOBS level of the IBM Input/Output Control System and by the FORTRAN input routines. It is intended that data in catalog format be in a useful form for as many users as possible.

Specialized input and output routines have been implemented to move catalog elements between core storage and magnetic tape. They perform all input and output operations in programs currently being written and planned for the future. These routines, the Catalog Input/Output System (CIOS), are written in the IBM MAP language for the 7044 but are designed so that they can also be called by FORTRAN routines. They use the IOOP level of the IBM Input/Output Control System. Section 3 describes these routines and their use.

## 2. CATALOGS ON MAGNETIC TAPE

A catalog is a body of data that can be mapped onto a tree in a certain way. There is a one-to-one correspondence between data of the catalog and nodes of the tree. Each datum is either a simple datum, or is itself a catalog. The information carried by a datum is its value. If a datum has no value, it is null.

A datum is a governor if there are branches to the next lower level emanating from the node to which it corresponds. A datum is dependent if it is on a level other than level one; data on level one are independent. Each governor has zero or more dependents with no limit set on the number possible; each dependent has only one governor. Although a datum on level one is independent, it is helpful to consider it a dependent of a pseudo element on level zero.

A catalog, consisting only of simple data, is a simple catalog. A catalog, with one or more data that are themselves catalogs, is a nesting catalog. A catalog that is a datum within a higher order catalog is a nested catalog.

The format adopted for catalogs on magnetic tape places no restriction on the amount of information, the number of data, or the extent of nesting in a catalog. A catalog is continued over as many reels of tape as necessary. Also, any number of independent catalogs can be written consecutively on tape. The physical reels of tape that contain a set of consecutive catalogs comprise a logical tape. The physical reels of a logical tape are written so that it can be read either as one continuous tape, or independent of one another. Each physical reel of a logical tape has a label that includes the name of the logical tape and the sequence number of the reel within the logical tape.

The tape format for catalogs also facilitates reading in a backward as well as a forward direction. Information is blocked into records that can vary in length but that do not exceed a limit set for each logical tape. The records of a physical tape are always followed by an end-of-file mark.

Some catalog applications are expected to produce many null data. The tape format permits null data to be deleted in the writing process. However, enough information is preserved so that the position of each can be recognized.

The data of a catalog are put into serial order by the following rule: A datum precedes those beneath it in the tree and those on the same level to the right. Thus the datum at the root of the tree is first, followed by those on its leftmost branch. The first datum on a branch precedes others on the branch; it is followed by the datum on its leftmost branch. If a datum is itself a catalog, that catalog is written in its entirety before continuing with the nesting catalog.

The first datum of a catalog is preceded by a copy of the map for the catalog and a begin-catalog marker. An end-catalog marker follows the last datum. Within the map, information for classes of data is put into serial order by the same rule used for a catalog datum.

## 2.1 A Blocking Format

Catalogs are collections of many, relatively small, information packages of variable length. For efficient storage on tape, each is treated as a logical record; logical records are packed into physical records or blocks. The user sets a limit on block size for each logical tape. A logical record can be divided between two or more blocks. This prevents the limit on block size from restricting the size of a logical record. It also permits blocks to carry the maximum amount of information.

The parts of a block are bound together by 36-bit linkwords. A linkword precedes the first, separates consecutive, and follows the last logical record or partial logical record in a block. The address portion of a linkword contains the number of 36-bit words between it and the next following one; the decrement contains the number of 36-bit words between it and the next preceding one. The decrement of the first linkword of a block and the address of the last are always 0. These two linkwords are also marked as block boundaries by a 1 in bit position two; all other linkwords have a 0 in position two. The last linkword of the last block on a physical tape is marked with a 1 in bit position one. The last linkword of every other block on the tape has a 0 in this position. If a logical record is continued from one block to another, the last linkword of the first block and the first linkword of the second block are each marked with a 1 in bit position twenty. In all other linkwords bit position twenty is 0.

The first two 36-bit words of a block are the IOBS\* control word and the FORTRAN control word, required to read the tapes with input routines from these two systems. The contents of these two control words are:

---

\*IOBS is the IOBS level of the IBM Input/Output Control System.

IOBS	5	n-1	2	00044
FORTTRAN	0	n-2	0	00001
	S 3		18 21	

The octal integer  $n$  is the number of 36-bit words in the block, and the quantities  $n-1$  and  $n-2$  are each 15 bits in length. These control words are followed by the linkwords and logical records of the block. Two blocks are diagramed in Fig. 1; a logical record is continued from the first to the second, and the second is the last block of a physical tape. Shaded portions of the linkwords are not used in the blocking format.

## 2.2 Catalogs and the Blocking Format

The blocking conventions described in Section 1.1 provide the basic format for catalog tapes. A label is written at the beginning of each reel to tie together the parts of a logical tape. The user can repeat selected information at the beginning of each reel to facilitate use of the reel independent of preceding reels of the logical tape. Tape labels, catalog maps, and simple data are written as logical records. Some of the unused bit positions in the linkwords are used to mark the beginning and end of catalogs and the beginning of data on each physical reel.

Tape Labels. A four-word tape label is the first logical record on each physical reel. The first word contains the limit set on block size for the logical tape.

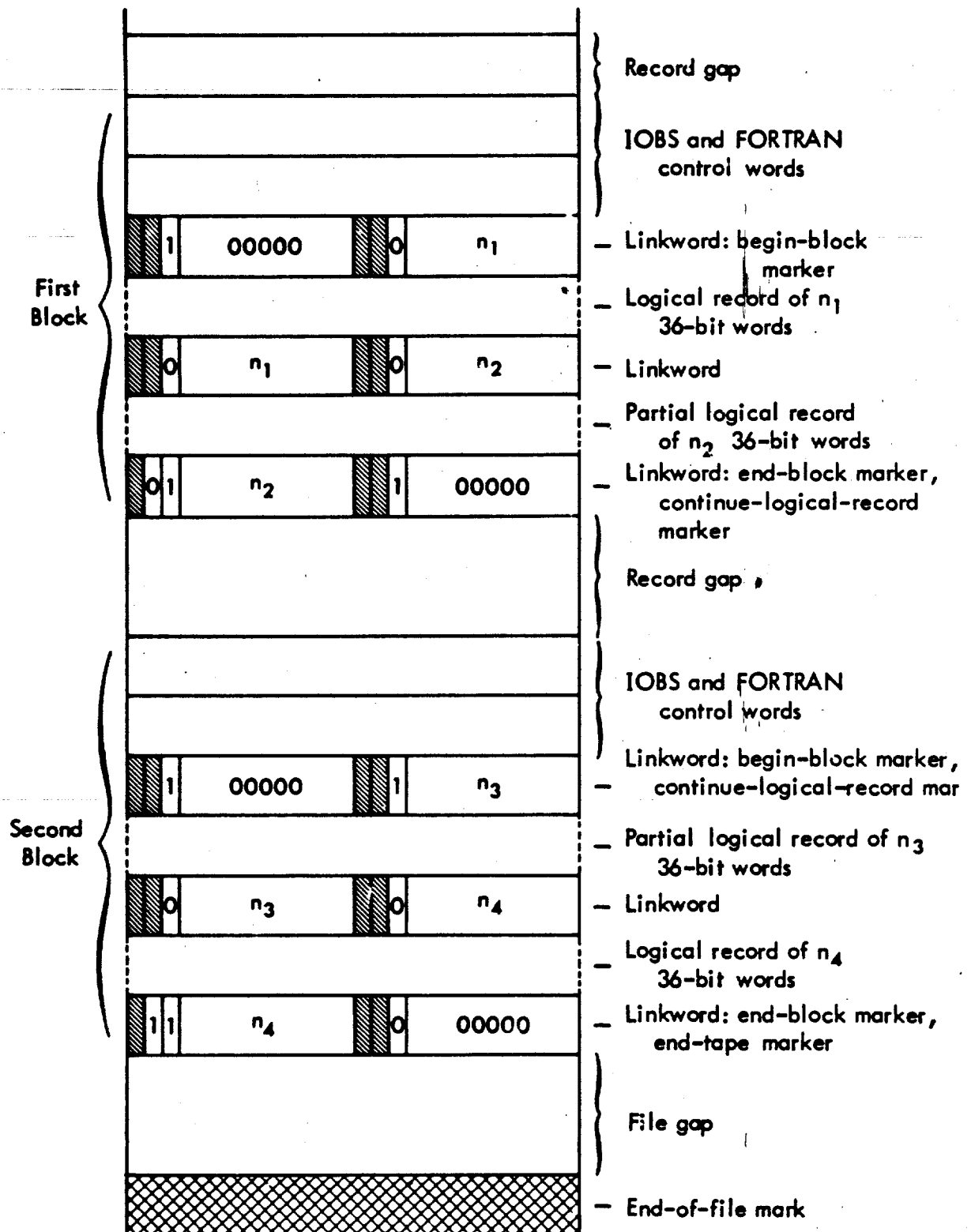


Fig.1 — Blocking Format

It is a 15-bit integer right-justified; the rest of the word is not used. The second word contains the date when the tape reel was written; there are two 6-bit integers each for month, day of the month, and year. The third word contains the logical tape name, the logical tape version number, and the sequence number of the physical reel within the logical tape. The tape name is three 6-bit characters; the version number consists of two 6-bit characters; the sequence number is a 6-bit integer. The fourth word of the label carries the number of repeated data at the beginning of the physical reel, written as a 15-bit integer right-justified. Its use is explained in the paragraphs Beginning a Physical Reel later in this section. A diagram of a tape label follows:

			Block Limit
Date			
Logical Tape Name		Tape Version Number	Reel Sequence Number
			Repeated Data Count

Catalog Maps. The map of a catalog contains one 36-bit word of information for each data class. These words are serially ordered by the same rule as is used to order



catalog elements (see Sec. 1.1). Each word has three parts: a class name, an encoding-type tag, and the level number for the class. The class name has three 6-bit characters; any combination of Hollerith characters can be used except three zeros.

The encoding-type tag for a class names the set of conventions used to encode the data of that class. The tag is a 6-bit character. Four encoding types and tags have been defined:

- B (bit pattern 010010) for elements written as 35-bit binary integers plus sign.
- H (bit pattern 011000) for elements written as a string of 6-bit Hollerith characters.
- R (bit pattern 101001) for elements encoded by the conventions of the RAND text encoding scheme.\*
- C (bit pattern 010011) for elements that are themselves catalogs.

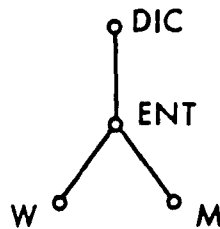
The level number for a class is the level on which the class occurs within the tree. The class at the root of the tree is on level one, its dependents are on level two, etc. The level number is a 9-bit integer. The information word for each data class will have 3 unused bits.

Associated with each datum of a catalog is the index to the position of its class within the map. The index for

---

\*This encoding scheme is described in M. Kay and T. Ziehe, Natural Language in Computer Form, The RAND Corporation, RM-4390-PR, February 1965.

the first class in a map is 0, for the second class is 1, etc. The following diagram, for example, gives the map and the index value for each class of a catalog:



Index	Class Name	Encoding Type	Level Number
0:	DIC	H	1
1:	ENT	B	2
2:	__W	R	3
3:	__M	R	3

5 18 24 27 Bit position

Data. Each datum of a catalog is accompanied on tape by a 36-bit control word. For a simple datum, the control word and the value of the datum are combined and written as a logical record. If the value is null, the CIOS routines drop the entire datum. In principle, however, a null datum can be written onto tape; it would be represented by its control word written as a logical record. A nested catalog is represented by its control word, written as a logical record, followed by the map and the data of the nested catalog.

A datum control word contains three items: an index to a class name in the map, the preceding implicit level (PIL) number for the element, and the level number for the class. Each item is a 9-bit integer; the first nine bits of the control word are not used. The index to the class name and the level number have just been explained; the level number is repeated in the control word only for convenience.

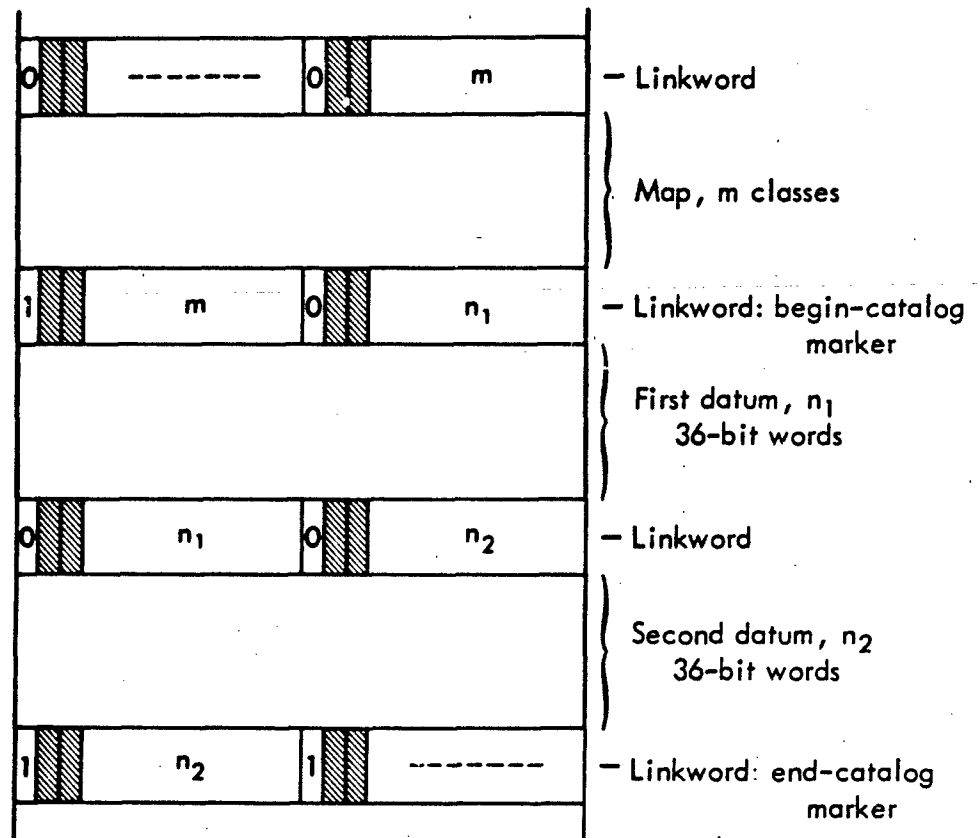
The PIL number for a datum is the number of the lowest (in numeric value) level touched by tracing a path through the catalog tree from the last preceding non-null datum to the current element. The PIL value for the first datum of a catalog is 0. When the governor of a datum is non-null or when a datum is not the first dependent of its governor, the element's PIL value is the level number of its governor. If an element's governor is null and it is the first dependent, its PIL value is the level number of the last non-null datum up the chain of governors.

The following diagram illustrates the format of a logical record for a non-null datum whose value requires three 36-bit words:

	Class Name Index	PIL Number	Level Number

} Value  
of the  
datum

Catalog Boundaries. The beginning and end of each catalog is explicitly marked. The linkword that precedes the first datum of a catalog carries the begin-catalog mark; the end-catalog mark is in the linkword that follows the last datum. These marks are stored in bit positions zero and eighteen. The pattern 1...0 equals begin-catalog, and 1...1 equals end-catalog. All other linkwords have the pattern 0...0 in these two positions. The pattern 0...1 is not defined. Each catalog on magnetic tape consists of a map, a begin-catalog mark, the elements of the catalog, and an end-catalog mark. The following diagram illustrates a single catalog with  $m$  classes and two data:



Begin- and end-catalog marks are assumed always to occur in pairs. Thus the boundaries of any catalog can be determined no matter what the pattern or extent of nesting may be. Certain operations provided in the system, notably CLOSE and SWITCH, will not always work properly if a change of tape occurs in the middle of a nested catalog.

Beginning a Physical Reel. Each physical reel of a logical tape is written so as to facilitate its use apart from preceding reels. After the label, the map of the catalog currently open is written. Following this the user has an option to repeat any number of catalog data from the

preceding reel or reels. The number of data repeated is recorded as the repeated data count in the label. When the reel is read as a continuation of the preceding reel, the logical records of repeated information are skipped. When a reel is read independent of the preceding reel, the map and repeated data serve as a bridge into the main body of data.

Because only one map is written at the beginning of a continuation reel, the map for the currently open catalog, a continuation reel cannot be used apart from the preceding reel if it begins with the map of a nested catalog. Therefore, this facility is useful only with simple catalogs.

To make the format for catalogs on magnetic tape functional for reading in a reverse direction, a begin-tape mark is placed in the linkword following the map at the beginning of a reel. This mark is a 1 in bit position one. However, no provision is made to recognize the boundary between new and repeated data when reading in the reverse direction. Nor is it possible to move directly from one reel of tape into the preceding one when reading in the reverse direction.

Co-occurrence of Begin-End Markers. Both the blocking format and conventions for writing catalogs into this format define markers that are recorded as bits in the prefix and tag of linkwords. In many cases two or more of these

markers can occur between two consecutive logical records. When possible, co-occurring markers are written into a single linkword. Whenever this would result in loss of information or ambiguity, consecutive linkwords are generated. Several conventions have been adopted that govern co-occurrence of markers and the use of consecutive linkwords.

Every block must have both a begin-block marker and an end-block marker. Since the same marker is used for both, each block must have at least two linkwords. Each physical tape must have both a begin-tape marker, required by the catalog conventions, and an end-tape marker, required by the blocking format. The same marker is used for both, but the latter can occur only in a linkword with the end-block marker. The former can occur in any linkword other than the last one of a block.

Since catalogs can contain other catalogs, multiple beginnings and multiple endings are possible. Consecutive catalog beginnings will not result in markers in the same linkword, but consecutive endings would. Instead, consecutive linkwords, one for each ending, are used to avoid the loss of information. Consecutive linkwords are also used when a catalog with no elements is written; the first linkword has the begin-catalog marker and the second the end marker. When a catalog begins immediately following a catalog end, the markers naturally fall into separate linkwords.

A final convention prevents the parts of a logical record from being written onto separate tape reels. This prevents the continued logical record marker from co-occurring with the begin-tape and end-tape markers. This marker can, therefore, only occur in a linkword that has either an end-block or begin-block marker.

The table in Appendix A shows all possible co-occurrences of markers and their meanings. Appendix B contains some diagrams that further illustrate the beginnings and endings of catalogs and tapes.

### 3. THE CATALOG INPUT/OUTPUT SYSTEM

The RAND Catalog Input/Output System (CIOS) allows the user to read or write catalogs on a set of predefined logical units. Each logical unit is normally assigned a primary and a secondary IBSYS utility unit. All input/output operations take place on the primary unit. When operations on the primary unit have been completed, the primary and secondary units can be switched and operations continued on the new primary unit. Information is read or written in standard catalog format as described in Section 2.

The Catalog Input/Output System consists of a set of subroutines that perform specific operations on the designated logical unit. The subroutines are entered by a CALL statement in either FORTRAN or MAP language.



The general form of a CALL statement is

CALL subroutine(S,E,p<sub>1</sub>...p<sub>n</sub>)

The parameters within the parentheses are S, the cell that contains the logical unit number; E, the cell into which return conditions are to be placed; and p<sub>1</sub>...p<sub>n</sub>, which are special parameters that may be required for the particular operation. The special parameters are listed and explained in the operation descriptions (Sec. 2.1).

Upon exit from the subroutine, cell E contains an integer that indicates whether the operation was successfully completed. For all operations the following standard return conditions are used:

- 0 Operation complete, no errors
- 1 Illegal unit designation
- 2 Illegal unit status

Condition 0 means that the operation was completed with no errors. Condition 1 means that the value in S is not an acceptable logical unit number. Condition 2 occurs if the initial status of the designated logical unit is illegal for the operation requested, for example, attempting to activate a unit that is already active or attempting to read a datum from an active write unit. If errors 1 or 2 occur, the requested operation is ignored. The unit status requirements are given under the operation descriptions and in Table 1.

Table 1  
CLOS OPERATIONS

<u>Level</u>	<u>Call</u>	<u>Operation</u>	<u>Initial Unit Status</u>
Control	ACTR(S,E,R1,B2,I)	Activate read unit	Inactive
	ACTW(S,E,R1,B2,I)	Activate write unit	Inactive
	INACT(S,E)	Inactivate unit	Active read/write
	SKIP(S,E)	Skip next catalog	Active read
	SKPP(S,E)	Skip previous catalog	Active read
III	SWITCH(S,E,R)	Switch units	Active read/write
	OPEN(S,E,M1,M2)	Open catalog	Active read/write
	CLOSE(S,E)	Close catalog	Active read/write
	GETD(S,E,D1,D2,D3)	Get next datum	Active read
	GEPD(S,E,D1,D2,D3)	Get previous datum	Active read
II	PUTD(S,E,D1,D2,D3)	Output datum	Active write
	GETL(S,E,L1,L2,L3)	Get next logical record	Active read
	GEPL(S,E,L1,L2,L3)	Get previous logical record	Active read
	PUTL(S,E,L1,L2,L3)	Output logical record	Active write
	GETB(S,E)	Get next block	Active read
I	GEPB(S,E)	Get previous block	Active read
	PUTB(S,E)	Output block	Active write
	REWIND(S,E)	Rewind tape	Active read/write
	REMOVE(S,E)	Remove tape	Active read/write

Special return conditions for a particular operation are given under the operation description. If a permanent tape error occurs while reading or writing, the job is terminated with an error message.

Three operation levels, plus a control level, are available in CIOS. Level III operations read and write in the datum mode, checking the data against the catalog map; Level II operations block and unblock logical records and process linkwords; Level I operations do the actual reading and writing of physical records (blocks) on tape and perform buffer switching. The Control commands initiate the logical unit for the desired mode of operation.

The user will normally use Level III and Control operations exclusively. Although Level II and Level I operations are made available to the user, they are normally used only by other system routines. To avoid errors, operations on a logical unit, once started on a particular level, should continue on that level.

Table 1 lists the available operations; Table 2 lists the CIOS return conditions.

### 3.1. Control Operations

To perform input/output operations on any level, the designated logical unit must first be activated. That is, buffers must be assigned, a control block must be set up, and the tape label must be processed.

Table 2  
CIOS RETURN CONDITIONS

<u>Operation</u>	<u>Condition</u>	<u>Meaning</u>
all	0	Operation complete, no errors
	1	Illegal unit designation
	2	Illegal unit status
ACTR	3	No tape label
	4	Inadequate buffer pool
	5	Reel sequence break
ACTW	4	Inadequate buffer pool
CLOSE	3	End of tape
	4	Catalog not open
GEPD	3	Beginning of tape
	4	Beginning of current catalog
	5	Illegal tape format
	6	Incomplete datum transmitted
	7	Catalog not open
	8	Catalog map incorrect
GEPL	3	Beginning of tape
	4	Incomplete record transmitted
GETB	3	End of file
GETD	3	End of tape
	4	End of current catalog
	5	Illegal tape format
	6	Incomplete datum transmitted
	7	Catalog not open
	8	Catalog map incorrect
GETL	3	End of tape
	4	Incomplete record transmitted
OPEN	3	End of tape
	4	Illegal tape format
	5	Inadequate map buffer
PUTB	3	End of tape

Table 2--Continued

<u>Operation</u>	<u>Condition</u>	<u>Meaning</u>
PUTD	3	End of tape
	4	Illegal data class name or index
	5	Data sequence error
PUTL	3	End of tape
SKIP	3	End of tape
	4	Attempt to skip open catalog
SKPP	3	Beginning of tape
	4	Attempt to skip open catalog
SWITCH	3	No alternate unit
		Label error on alternate unit

Buffers and a catalog control block (CCB) are assigned from a buffer pool provided by the user. The buffer pool is a block of storage whose size must be at least  $(2N+15)$  words for each logical unit activated, where  $N$  is the block size for the tape being read or written.

The catalog control block is set up at the beginning of the buffer pool area and contains all information pertaining to a logical unit and the catalogs that are being processed on that logical unit. The format of a CCB is shown in Fig. 2.

The bits in the CBSNS word of the CCB are listed in Table 3. During operation, CIOS maintains the sense bits as follows: if the stated condition is present, the bit is set to 1; if the stated condition is absent, the bit is reset to 0.

After the control block has been set up, the remaining words in the buffer pool, less those required as IOOP control words, are divided into two buffers of equal length  $N$ . One buffer is designated as the current buffer and the other as alternate. At all times, actual transmission of data between core and tape takes place in the alternate buffer, while the current buffer is available to the user. Buffer switching is performed on Level I.

3.1.1. Activate Read Unit. A control block and buffers are set up, the tape on the designated logical unit is rewound, and the first block is read into the current

CBUNI		Address of alternate unit		Address of primary unit
CBBUF		Number of data words in current buffer		Starting location of current buffer
CBBLK		Block size		Starting location of alternate buffer
CBCUR				Location of current linkword
CBLBL		Logical tape name	Tape version number	Reel sequence number
CBMAP		Size of current map		Starting location of current map
CBSNS	P R	P B S L L C	R N O C	R R B B B E E E E C N B C T B C N T
CBCWD		Class name index	PIL number	Level number

Fig.2-- Catalog control block

Table 3  
CCB SENSE BITS

<u>Bit</u>	<u>Name</u>	<u>Condition</u>
S	PR	Current unit is primary
1	PB	Buffers are primed
2	BS	Beginning of tape has been sensed
3	RL	Reading reverse, null data are being transmitted
4	NL	Null data are being transmitted
5	OC	An open catalog request is present
27	RC	Reading reverse, linkword pointer is at end of catalog
28	RN	Reading reverse, linkword pointer is at end of nested catalog
29	BB	Linkword pointer is at beginning of block
30	BC	Linkword pointer is at beginning of catalog
31	BT	Linkword pointer is at beginning of tape
32	EB	Linkword pointer is at end of block
33	EC	Linkword pointer is at end of catalog
34	EN	Linkword pointer is at end of nested catalog
35	ET	Linkword pointer is at end of tape



buffer. The tape label is checked and transmitted to a label array provided by the user, and the logical unit is then ready to perform read operations on any level. The status of the designated logical unit must initially be inactive.

The calling sequence is

CALL ACTR(S,E,B1,B2,I)

The special parameters are B1, starting location of the buffer pool; B2, a cell containing the size of the buffer pool; and I, starting location of the label array.

The label array is a six-word block into which the tape label is read. The information is placed into the array right-justified and ordered as follows: I(1), tape name; I(2), version number; I(3), reel sequence number; I(4), repeated data count; I(5), block size; I(6), date.

A special option is provided by the calling sequence

CALL ACTR(S,E,B1,B2)

If the parameter I is null (not given), tape label information is not transmitted to the user, and a label array need not be provided.

The special return conditions are

- 3 No tape label
- 4 Inadequate buffer pool
- 5 Reel sequence break

If the first logical record on tape is not the standard four-word label or if the block size specified on the tape label is larger than the available buffer, the appropriate condition code is set, and activation is terminated. If the first reel mounted on the logical unit does not have a sequence number of one, activation is completed, and condition 5 is set as a warning.

3.1.2. Activate Write Unit. A control block and buffers are set up, and the tape on the designated logical unit is rewound. A label is formed from the information in the label array and is transmitted to the current output buffer. The unit is now ready to perform write operations on any level. The status of the designated logical unit must initially be inactive.

The calling sequence is

CALL ACTW(S,E,B1,B2,I)

The special parameters are B1, starting location of the buffer pool; B2, a cell containing the size of the buffer pool; and I, starting location of the label array.

The label array is a four-word block from which a tape label is formed. The information in the array must be right-justified and ordered as follows: I(1), tape name; I(2), version number; I(3), reel sequence number; I(4), repeated data count. If the reel sequence number word, I(3), contains zero, a sequence number of one will be generated by the system.

A special option is provided by the calling sequence

CALL ACTW(S,E,B1,B2)

If the parameter I is null (not given), a tape label is produced in which the tape name, version number, and repeated data count are all zeros and the reel sequence number is one.

The special return condition is

### 3 Inadequate buffer pool

If the buffer pool is not large enough to allow proper processing of the data, condition 3 is set, and activation is terminated. The minimum permissible size for a buffer pool is 25 words.

3.1.3. Inactivate Unit. The physical tape is rewound, the buffers and control block are released, and no further operations can be carried out on the unit. The status of the designated logical unit must initially be active read or active write.

The calling sequence is

CALL INACT(S,E)

If the unit is in active write status, an end of tape bit is set in the current linkword, the current buffer is truncated and written out, and an end of file mark is written on the tape.

3.1.4 Switch Units. The current tape on the designated logical unit is closed (see below) and unloaded. The primary and secondary units are interchanged. The new tape is rewound, and its label is processed. If the secondary unit is the same as the primary unit, the effect will be to continue operation on the same unit but on a new reel of tape. The status of the designated logical unit must initially be active read or active write.

The calling sequence is

CALL SWITCH(S,E,R)

The special parameter, R, is a word containing the repeated data count.

When reading, the tape name and version number in the new tape label are read and verified against those from the previous tape; the reel sequence number is checked for proper sequence. The sequence number on the new tape must be one greater than that on the previous tape. The repeated data and catalog map are skipped over, and the repeated data count is placed in R.

When writing, the current tape is closed as follows: The end of tape bit is set in the current linkword, the current buffer is truncated and written out, and an end of file mark is written. For the new tape, a label is produced identical to that written on the previous reel except that the reel sequence number is increased by one

and the repeated data count is taken from R. Following the label, the current map is output with the beginning of tape bit set in the trailing linkword.

The special conditions are

- 3 No alternate unit
- 4 Label error on alternate unit

For both reading and writing, if no alternate unit has been defined, no switching takes place, and an error condition is set. When reading, if there is no label or if label verification on the new tape fails, the operation is terminated, but the units will have already been switched.

3.1.5 Skip Next Catalog. The next complete catalog on tape is skipped over, and the logical record pointer is positioned to read the next logical record following the catalog. The catalog to be skipped must not be opened. The status of the designated logical unit must initially be active read.

The calling sequence is

CALL SKIP(S,E)

If a tape contains several sequential catalogs, this operation can be used to skip over those catalogs not to be read. One call to the SKIP subroutine will skip over one catalog, including any nested catalogs contained within the catalog being skipped.

Skipping should not be used in place of closing to terminate operations on the current catalog.

If skipping is initiated on an open catalog, the logical record pointer is positioned to the end of the catalog, but the catalog map remains in control, the end of catalog bit in the CCB remains set at "1", and an error return condition is set.

The special return conditions are

- 3 End of tape
- 4 Attempt to skip open catalog

If the end of tape is encountered or an end of catalog is detected before any beginning of catalog is detected, skipping is terminated and the appropriate condition set. If skipping terminates, the linkword pointer remains directed at the end of catalog or end of tape.

3.1.6. Skip Previous Catalog. The previous catalog on tape is skipped over, from back to front, and the logical record pointer is positioned at the leading linkword of the map for this catalog. The catalog to be skipped must not be opened, nor can it be the first catalog on tape. The status of the designated logical unit must initially be active read.

The calling sequence is

CALL SKPP(S,E)

One call to SKPP will skip over one catalog, including any nested catalogs contained within the catalog being skipped.

The special return conditions are

- 3 Beginning of tape
- 4 Attempt to skip open catalog

If the beginning of tape is detected or a beginning of catalog is detected before any end of catalog marks have been sensed, skipping is terminated and the appropriate condition set. If skipping terminates, the linkword pointer remains directed at the beginning of catalog or beginning of tape.

### 3.2. Level III Operations

Level III operations read and write in the datum mode. Information in the datum control word is interpreted in accordance with the map for the current catalog. Before operations can begin on the data within a catalog, that catalog must be opened so that its map is available to the system.

When reading, forward or reverse, CIOS verifies the level number of each datum against the level number of the corresponding map entry. If a read forward operation (GETD) is immediately followed by a read reverse operation (GEPD), or vice-versa, the same datum is transmitted to the user both times.

When writing, CIOS checks each datum against the immediately preceding datum to ensure that the data class and level number sequence is properly defined according to the map structure. The rules used for checking are given below.

If the level of the current data class is greater than that of the previous data class:

- (a) The difference in levels must equal one
- (b) The index of the current data class must be greater than that of the previous data class
- (c) In the map, any data class that lies between the previous data class and the current data class must have a level equal to or greater than that of the current data class.

If the level of the current data class is equal to that of the previous data class, then rule (c), above, applies and:

The index of the current data class must be equal to or greater than that of the previous data class.

If the level of the current data class is less than that of the previous data class and the index of the current data class is greater than that of the previous data class, then rule (c), above, applies.

If the level of the current data class is less than that of the previous data class and the index of the current data class is less than that of the previous data class then:



In the map, any data class that lies between the previous data class and the current data class must have a level greater than that of the current data class.

3.2.1. Open Catalog. The location and size of the specified map are placed in the CCB, and that map is used for all further input/output operations on the logical unit. The status of the designated logical unit must initially be active read or active write.

The calling sequence is

CALL OPEN(S,E,M1,M2)

The special parameters are M1, starting location of the map array; and M2, a word containing the size of the map array.

Upon completion of the operation, the accumulator contains, as an integer, the number of data classes in the map.

The map array is a block of words into which the catalog map is to be read or from which the catalog map is to be written. The size of the array must be at least  $(C+2)$  words, where  $C$  is the number of data classes in the map. This value,  $(C+2)$ , must also appear in M2. Two extra words are required at the end of the map for system linkages as described below.

When reading, the catalog map is transmitted to the map array. Initially, the map must be the next logical record to be read by the system.

When writing, the catalog map is transmitted from the array to the output buffer followed by a trailing linkword

in which the beginning of the catalog bit is set. If the catalog being opened is the first catalog on tape, the beginning of tape bit is also set in the trailing linkword.

When a nested catalog is opened, the location and size of the map of the nesting catalog are stored at the end of the map for the nested catalog. The map for the nested catalog is used for all further operations until the nested catalog is closed. When the nested catalog is closed, the map for the nesting catalog is restored, and operations can be resumed on that catalog.

The special return conditions are

- 3 End of tape
- 4 Illegal tape format
- 5 Inadequate map array

When reading, if the end of tape is detected, opening is not completed. To continue, it is necessary to switch reels and open the new reel. If the logical record pointer is not positioned in front of the map or if an attempt is made to open a nested catalog that has not been preceded by the proper datum control word, condition 4 is set in E, and opening is terminated. If the map array is inadequate to contain the catalog map, opening is terminated.

When writing, if the end of tape is detected, opening is completed, and to continue, it is only necessary to switch reels. If an attempt is made to open a nested catalog before

the proper datum control word has been written, condition 4 is set, and opening is not performed.

In all cases, if opening is not completed, the previous catalog map, if any, remains in control, and the map parameters, M1 and M2, are ignored.

3.2.2. Close Catalog. The map for the current catalog is released, and the end of catalog bit in the CCB is reset. No further operations can be performed on the catalog. The status of the designated logical unit must initially be active read or active write.

The calling sequence is

CALL CLOSE(S,E)

If the catalog being closed is a nested catalog, the map for the nesting catalog, of which the nested one is an element, is set in control. Operations can now be resumed on the nesting catalog.

When reading, the tape is positioned to the end of the current catalog.

When writing, the end of catalog bits are set in the current linkword.

If no catalog is open on the designated logical unit, condition 4 (catalog not open) is set, and the operation is ignored. If the end of tape is detected before the end of catalog, condition 3 is set, and to complete the operation the user must call SWITCH and then call CLOSE again. However, if the catalog being closed contains a

nested catalog in the middle of which an end of tape occurs, then the close operation cannot be successfully completed.

3.2.3. Get Next Datum. The next datum of the current catalog is transmitted into the datum array, and the data class index is put in the designated location. If the end of catalog has been reached, no information is transmitted, and a special return condition is set. The status of the designated logical unit must initially be active read.

The calling sequence is

CALL GETD(S,E,D1,D2,D3)

The special parameters are D1, starting location of the datum array; D2, a cell containing the size of the datum array; and D3, a cell into which the data class index is to be placed as an integer.

Upon completion of the operation, the accumulator contains the actual number of words transmitted into the datum array.

If the current datum control word indicates one or more null data precede the current datum, these are presented to the user one at a time on successive calls. The logical record pointer will not move forward until all null data have been sent. A null datum is indicated to the user by a word count of zero in the accumulator.

If the datum control word indicates that a nested catalog follows, two options are possible. To ignore the catalog, another GETD command must be given. The catalog will be skipped, and the datum immediately following the catalog will be transmitted into the datum array. To read data from the nested catalog, an OPEN command must be given.

If the datum array is not large enough to accommodate the entire datum, as much as possible is transmitted, a special return condition is set into E, and the accumulator contains the number of words that were actually transmitted into the array. The remainder of the datum is skipped over.

A special option is provided by the calling sequence

CALL GETD(S,E)

If the parameters D1, D2, and D3 are null (not given), no information is transmitted into the datum array, but the next datum is skipped over, and the accumulator is set to contain the size of the skipped datum.

The return conditions are

- 3 End of tape
- 4 End of current catalog
- 5 Unable to skip nested catalog (illegal tape format)
- 6 Incomplete datum transmitted
- 7 Catalog not open
- 8 Catalog map incorrect

If the end of tape is detected, it is necessary to switch reels and repeat the GETD operation to continue. If the end

of catalog is detected, it is necessary to close the current catalog before operations can continue. If a nested catalog is closed, operations can immediately be resumed on the nesting catalog.

When a datum is read, the index in the datum control word is used to locate the corresponding data class in the map. The level in the datum control word is checked against the level of the map entry. If the two disagree, condition 8 is set in E. Condition 8 also results if CIOS is unable to locate, in the catalog map, a data class of the proper level to satisfy a null datum.

If conditions 3, 4, 5, 7, or 8 occur, no information has been transmitted into the datum array.

3.2.4. Get Previous Datum. The previous datum of the current catalog is transmitted into the datum array, and the data class index is put in the designated location. If the beginning of catalog is reached, no information is transmitted, and a special return condition is set. The status of the designated logical unit must initially be active read.

The calling sequence is

CALL GEPD(S,E,D1,D2,D3)

The special parameters are D1, starting location of the datum array; D2, a cell containing the size of the datum array; and D3, a cell into which the data class index is to be placed as an integer.

Upon completion of the operation, the accumulator contains the actual number of words transmitted into the datum array.

If the current datum control word indicates one or more null data follow transmission of the current datum, these are presented to the user one at a time on successive calls. The logical record pointer will not back up until all null data have been sent. A null datum is indicated to the user by a word count of zero in the accumulator.

If a nested catalog is encountered while reading in the reverse direction, it will be skipped over and its datum control word transmitted to the user. It is not possible to read into the nested catalog directly. To read the nested catalog, the user must re-read the datum control word in a forward direction and then open the catalog.

If the datum array is not large enough to accommodate the entire datum, as much as possible is transmitted, a special return condition is set, and the accumulator contains the number of words that were actually transmitted. The remainder of the datum is ignored.

A special option is provided by the calling sequence

CALL GEPD(S,E)

If the parameters D1, D2, and D3 are null (not given), no information is transmitted into the datum array, but the

previous datum is skipped over, and the accumulator is set to contain the size of the skipped datum.

The return conditions are

- 3 Beginning of tape
- 4 Beginning of current catalog
- 5 Illegal tape format
- 6 Incomplete datum transmitted
- 7 Catalog not open
- 8 Catalog map incorrect

If the beginning of tape is detected, it is not possible to continue by simply switching reels since the alternate reel will be positioned at the start. When a datum is read, the index in the datum control word is used to locate the corresponding data class in the map. The level in the datum control word is checked against the level of the map entry. If the two disagree, condition 8 is set. Condition 8 also results if CIOS is unable to locate, in the catalog map, a data class of the proper level to satisfy a null datum.

If conditions 3, 4, 5, 7, or 8 occur, no information has been transmitted into the datum array.

3.2.5. Output Datum. The information from the datum array is transmitted to the output buffer, preceded by a datum control word formed from the information in the catalog map. The status of the designated logical unit must initially be active write.



The calling sequence is

CALL PUTD(S,E,D1,D2,D3)

The special parameters are D1, the starting location of the datum array; D2, a cell containing the size of the datum to be output; and D3, a cell containing the data class name or data class index.

The contents of D3 can either be the data class index as a right-justified integer, or the data class name in the leftmost 18 bits of the word. If the data class name is given, its format must be the same as it appears in the catalog map; the system computes the appropriate index.

Sequence checking is performed as described previously.

If the data class to be written is a nested catalog, its control word must first be output as a datum of length zero; the nested catalog must then be opened. When the elements of the nested catalog have all been written, a CLOSE operation will allow writing to resume on the nesting catalog. If the nested catalog is to be null, only its control word need be output.

The special return conditions are:

- 3 End of tape
- 4 Illegal data class name or index
- 5 Data sequence error

If end of tape is reached (condition 3), the datum has already been placed in the output buffer and need not be repeated. The user has the options of closing the catalog,

inactivating the unit, or switching reels to continue the current catalog. Condition 4 means either that the data class name is not in the map or that the data class index lies outside the range of the map. If the datum fails any of the sequence checks, condition 5 is given, and the datum is ignored.

### 3.3. Level II Operations

Level II operations process logical records in the current buffer. A logical record is defined as the information contained between two successive linkwords, neither of which is marked as intermediate. The linkword that immediately precedes a logical record is called its leading linkword, while the linkword that immediately follows a logical record is called its trailing linkword. If a logical record continues over more than one physical record, it may also have associated with it intermediate linkwords.

When reading, Level II transmits logical records from the current buffer to the user, one at a time, under control of a logical record pointer maintained in word CBCUR of the CCB.

In a read forward operation (GETL), the pointer is initially directed at the leading linkword of the next logical record. After the logical record has been transmitted to the user, the pointer is directed at its trailing linkword. If the trailing linkword is also the end of the

block, a call is immediately set up to Level I (GETB) to get the next block and direct the pointer to the first linkword of the block. If the end of tape has been reached, no information is transmitted, and a special return condition is set.

In a read reverse operation (GEPL), the pointer is initially directed at the trailing linkword of the desired logical record. After the logical record has been transmitted to the user, the pointer is directed at its leading linkword. If the pointer is initially directed at the first linkword of the block, a call is immediately set up to Level I (GEPB) to get the previous block and direct the pointer to the last linkword of the block. If the beginning of tape has been reached, no information is transmitted, and a special return condition is set.

If a read forward operation is immediately followed by a read reverse operation, or vice-versa, the same logical record is transmitted to the user both times.

Upon completion of any read operation, the linkword at which the logical record pointer is directed is checked for markers in the prefix and tag. If a marker bit is present, the appropriate flag bit is set in word CBSNS of the CCB and can be tested by the user. With the exception of ET (end of tape), all marker flags in the CCB are reset at the beginning of each read operation.

When writing, Level II puts logical records into the current buffer and maintains a pointer to the current linkword

plus a count of the number of data words in the current buffer. When the current buffer is full, a call is made to PUTB to switch buffers and output the alternate buffer.

3.3.1. Get Next Logical Record. The next logical record in the current buffer is transmitted into the logical record array; the datum control word is transmitted into the designated cell. The status of the designated logical unit must initially be active read.

The calling sequence is

CALL GETL(S,E,L1,L2,L3)

The special parameters are L1, starting location of the logical record array; L2, a cell containing the size of the logical record array; and L3, a cell into which the datum control word is to be placed.

Upon completion of the operation, the accumulator contains the number of words that were transmitted into the logical record array.

The logical record array must be large enough to accommodate the body of the logical record. If the array is not large enough, as many words as possible are transmitted, and a special return condition is set.

A special option is provided by the calling sequence

CALL GETL(S,E,L1,L2)

If the parameter L3 is null (not given), the logical record is assumed to have no datum control word and is transmitted to the user in its entirety.

Another special option is provided by the calling sequence

CALL GETL(S,E)

If the parameters L1, L2, and L3 are all null, no data is transmitted to the user, but the pointer is moved forward one logical record, and any markers found in the trailing link-word set the appropriate bits in the CCB. Upon completion of the operation, the accumulator contains the number of words in the entire logical record skipped over.

The special return conditions are

- 3 End of tape
- 4 Incomplete record transmitted

If the end of tape is detected, no information is transmitted.

3.3.2. Get Previous Logical Record. The previous logical record in the current buffer is transmitted into the logical record array; the datum control word is transmitted into the designated cell. The status of the designated logical unit must initially be active read.

The calling sequence is

CALL GEPL(S,E,L1,L2,L3)

The special parameters are L1, starting location of the logical record array; L2, a cell containing the size of the logical record array; and L3, a cell into which the datum control word is to be placed.

Upon completion of the operation, the accumulator contains the number of words that were transmitted into the logical record array.

The logical record array must be large enough to accommodate the body of the logical record. If the array is not large enough, as many words as possible are transmitted, and a special return condition is set.

A special option is provided by the calling sequence

CALL GEPL(S,E)

If the parameters L1, L2, and L3 are all null, no data is transmitted to the user, but the pointer is moved back one logical record, and any markers found in the leading linkword set the appropriate bits in the CCB. Upon completion of the operation, the accumulator contains the number of words in the entire logical record skipped over.

The special return conditions are

- 3 Beginning of tape
- 4 Incomplete record transmitted

If the beginning of tape is detected, no information is transmitted into the logical record array. If the logical

record array is not big enough to contain the entire logical record, as much as possible is transmitted, and the remainder of the logical record is ignored.

3.3.3. Output Logical Record. The information in the logical record array is put into the current buffer, preceded by the datum control word. If there is not enough room in the current buffer to put the entire logical record, as much as possible is put into the current buffer, the buffers are switched, and the remainder of the logical record is put into the new buffer. A logical record can be continued over as many physical records as necessary. The status of the designated logical unit must initially be active write.

The calling sequence is

CALL PUTL(S,E,L1,L2,L3)

The special parameters are L1, starting location of the logical record; L2, a cell containing the number of words in the body of the logical record; and L3, a cell containing the datum control word for the logical record.

The datum control word is placed at the front of the logical record. If it is necessary to continue the logical record over more than one physical record, the datum control word is not repeated.

A special option is provided by the calling sequence

CALL PUTL(S,E,L1,L2)

If the parameter L3 is null (not given), the logical record is assumed to have no datum control word, and only the number of words given in L2 is transmitted to the output buffer.

The special return condition is

### 3 End of tape

If the end of tape is reached, the logical record on that tape has already been placed in the output buffer and need not be repeated. At this time the user has the options of closing the current reel and switching reels, inactivating the logical unit, or closing the catalog.

#### 3.4. Level I Operations

Level I operations perform the actual tape reading, writing, and repositioning and are normally used only by other system routines.

All input and output is done using the initiate mode of synchronization available in the IOOP level of the IBSYS Input/Output Control System. This provides maximum efficiency of operation by allowing computation to proceed simultaneously with the actual reading and writing of tape.

3.4.1 Get Next Block. The buffers are switched and reading initiated into the alternate buffer. The number of data words in the current buffer is taken from the IOBS control word and set into the CCB. The logical record pointer



is set to the start of data in the current buffer. The status of the designated logical unit must initially be active read.

The calling sequence is

CALL GETB(S,E)

If an end of file is detected, return condition 3 is set, buffers are not switched, and no reading is initiated.

3.4.2. Get Previous Block. The tape is backspaced three physical records and reading initiated into the alternate buffer. When the alternate buffer is full, the buffers are switched and reading initiated into the new alternate buffer. The number of data words in the current buffer is taken from the IOBS control word and set into the CCB. The logical record pointer is set to the last link-word of the current buffer. The status of the designated logical unit must initially be active read.

The calling sequence is

CALL GEPB(S,E)

3.4.3 Output Block. An IOBS and a FORTRAN control word are generated and placed in front of the data in the current output buffer. When the previous write operation on the designated logical unit has been completed, buffers are switched, and writing is initiated from the alternate buffer. The data word count for the current buffer is set

to zero, and the pointer is directed to the start of data in the current buffer. The status of the designated logical unit must initially be active write.

The calling sequence is

CALL PUTB(S,E)

If the end of tape is sensed, condition 3 is transmitted to the user at the completion of the output operation. Writing can continue for a limited time on the designated logical unit, enabling the user to switch reels, or to inactivate the logical unit.

If a permanent write error is detected, the job is terminated.

3.4.4. Rewind Tape. The tape mounted on the designated logical unit is rewound. The status of the designated logical unit must initially be active read or active write.

The calling sequence is

CALL REWIND(S,E)

3.4.5. Remove Tape. The tape mounted on the designated logical unit is rewound and unloaded. The status of the designated logical unit must initially be active read or active write.

The calling sequence is

CALL REMOVE(S,E)

### 3.5. Logical Unit Table

The table of CIOS logical units is maintained as a separate MAP-coded program with a deck name "LTB." By reassembling this deck, the IBSYS utility units assigned to each CIOS logical unit can be changed.

Each inactive unit has an entry in the logical units table of the form

	Address of secondary unit		Address of primary unit
S	3	18	21

Each active unit has an entry in the logical units table of the form

AA			Pointer to the CCB
RW			
S1	3	18	21

Either AR is set (1) if the status of the unit is active read, or AW is set (1) if the status of the unit is active write. The address (bits 21-35) contains the 2's complement of the first location of the CCB.

When a unit is inactivated, its entry in the logical units table is restored so that the address contains the address of the primary unit, and the decrement (bits 3-17) contains the address of the secondary unit.

3.5.1. LTB Program Deck. Four types of entries are possible in the logical units table

- (a) PZE pri,,sec
- (b) PZE pri,,pri
- (c) PZE pri
- (d) PZE

Where pri refers to the primary unit and sec refers to the secondary unit, both are expressed in the form S.SUxx. Forms (a) and (b) allow reel switching; form (c) does not. Form (d) is used if no operations are to be carried out on a particular logical unit, and therefore nothing is to be attached to that unit.

The position of an entry in the logical units table corresponds to the logical unit number, with the numbering starting at 0. An entry must be present for each logical unit, up to the highest logical unit to be used. The highest logical unit number is one less than the number of entries in the table.

An example of a MAP-coded logical units table is shown below.

In the example, the highest logical unit number available is 5. Although logical units 2 and 3 cannot be used since nothing is attached to them, an entry must be present in the table for each unit. For logical units 1 and 5, the primary and secondary units are the same; while for logical unit 4, no secondary unit is defined.

<u>Name</u>	<u>Operation</u>	<u>Variable</u>	<u>Remarks</u>
\$IBMAP	LTR		
	ENTRY	LTORG	
	ENTRY	LTEND	
LTORG	BSS	0	START OF LOGICAL UNITS TABLE
	PZE	S.SU00,,S.SU01	LOGICAL UNIT 0
	PZE	S.SU04,,S.SU04	LOGICAL UNIT 1
	PZE		LOGICAL UNIT 2
	PZE		LOGICAL UNIT 3
	PZE	S.SU02	LOGICAL UNIT 4
	PZE	S.SU03,,S.SU03	LOGICAL UNIT 5
LTEND	BSS	0	END OF LOGICAL UNITS TABLE
	END		

The symbols LTRG and LTEND are entry points that must be defined as shown to indicate the size of the logical units table.

Appendix A

CO-OCCURRENCE OF BEGIN AND END MARKERS

Permitted Marker Patterns

Prefix	Tag*	Meaning
000	0-0	no markers
100	0-0	begin catalog
100	1-0	end catalog
010	0-0	begin tape
001	0-0	begin block, or end block
101	0-0	begin catalog and begin block, or begin catalog and end block
101	1-0	end catalog and begin block, or end catalog and end block
110	0-0	begin catalog and begin tape
110	1-0	end catalog and begin tape
011	0-0	begin tape and begin block, or begin tape and end block, or end tape and end block
111	0-0	begin catalog and begin tape and begin block, or begin catalog and begin tape and end block, or begin catalog and end tape and end block
111	1-0	end catalog and begin tape and begin block, or end catalog and begin tape and end block, or end catalog and end tape and end block
001	0-1	end block and continue logical record, or begin block and continue logical record.

\*The middle position of the tag field is not used.

Illegal Marker Patterns

Prefix	Tag <sup>*</sup>	Reason
100	0-1	the continue-logical-record marker cannot occur with either the begin-catalog or end-catalog marker
100	1-1	
101	0-1	
101	1-1	
110	0-1	
110	1-1	
111	0-1	
111	1-1	
010	0-1	the continue-logical-record marker cannot occur with the begin-tape (end-tape) marker
011	0-1	
000	0-1	the continue-logical-record marker cannot occur without the begin-block (end-block) marker
000	1-0	the bit pattern 0.. 1.. is not defined
000	1-1	
001	1-0	
001	1-1	
010	1-0	
010	1-1	
011	1-0	
011	1-1	

\*The middle position of the tag field is not used.

Appendix B

EXAMPLES OF BEGINNINGS AND ENDS  
OF CATALOGS AND TAPEC

ec



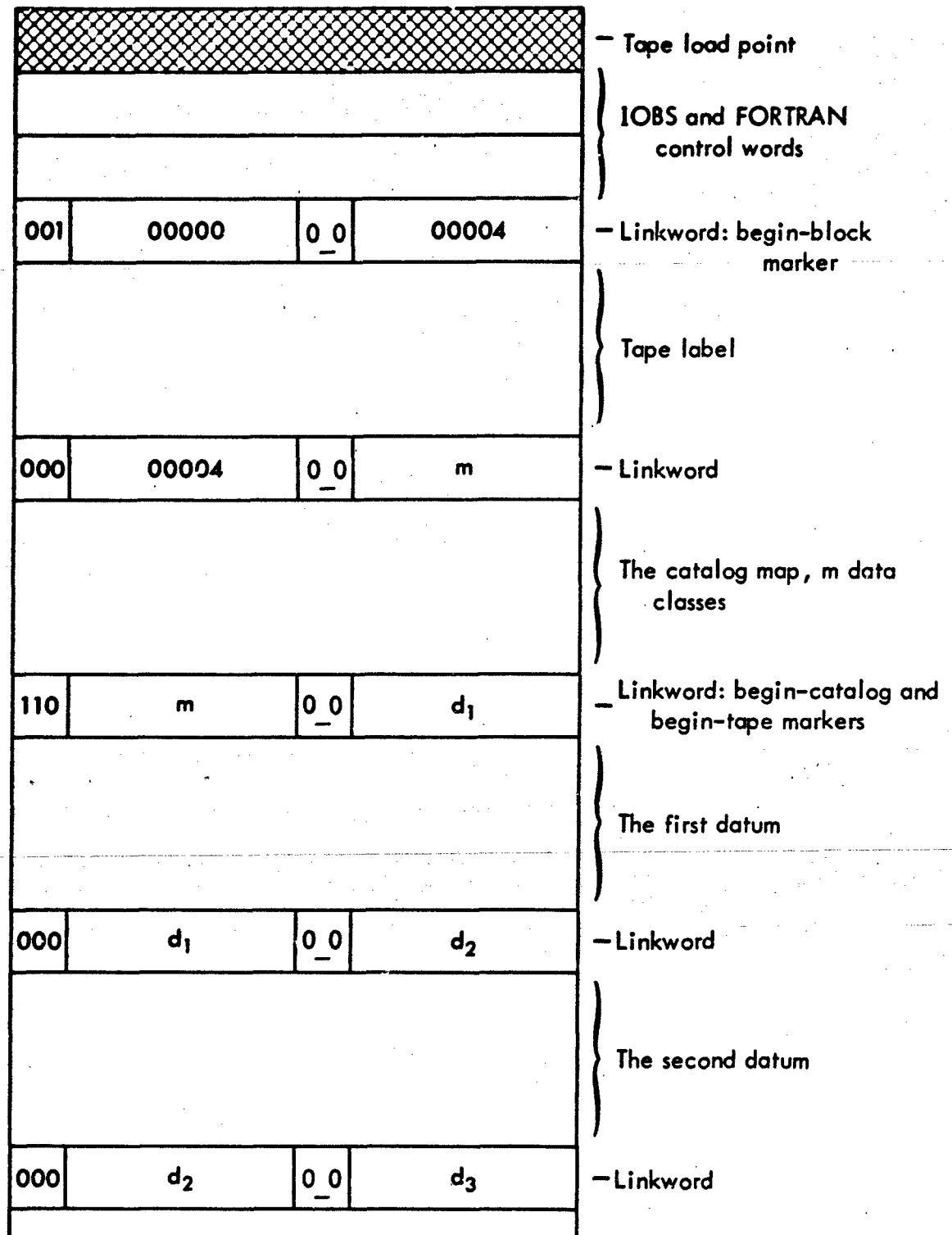


Fig.3 — Beginning of a catalog, the first of a logical tape

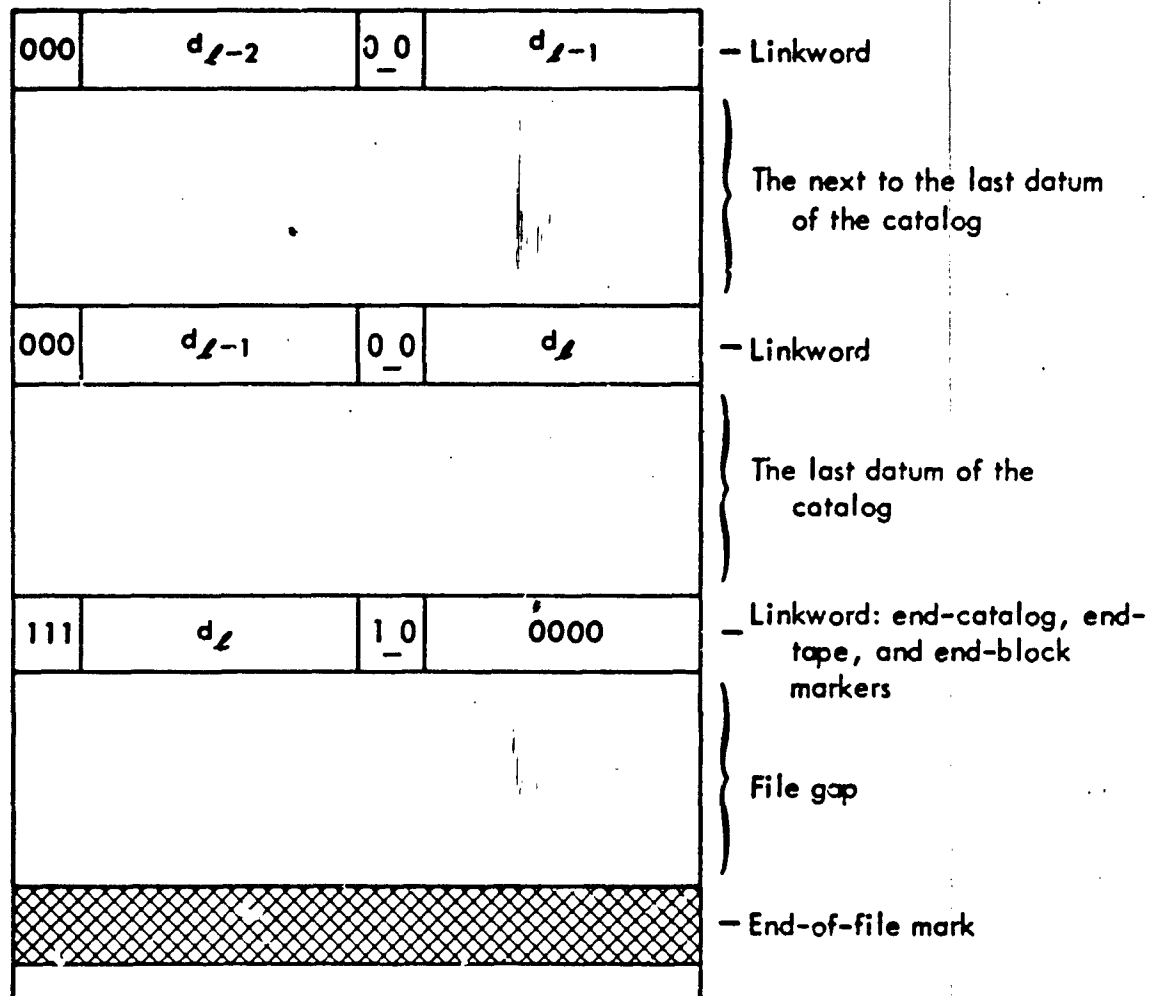


Fig.4— End of a catalog, the last of a logical tape

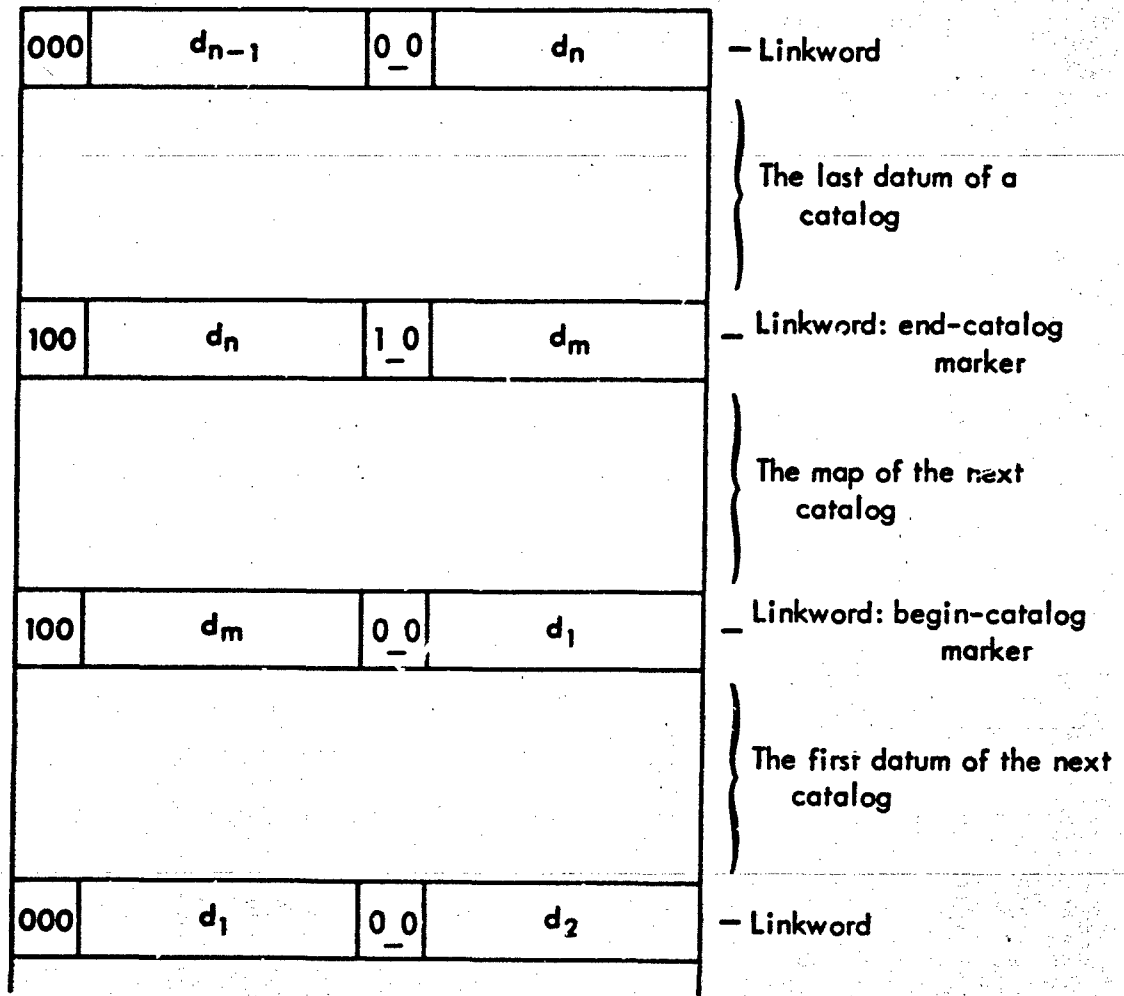


Fig.5— Interface of two independent catalogs

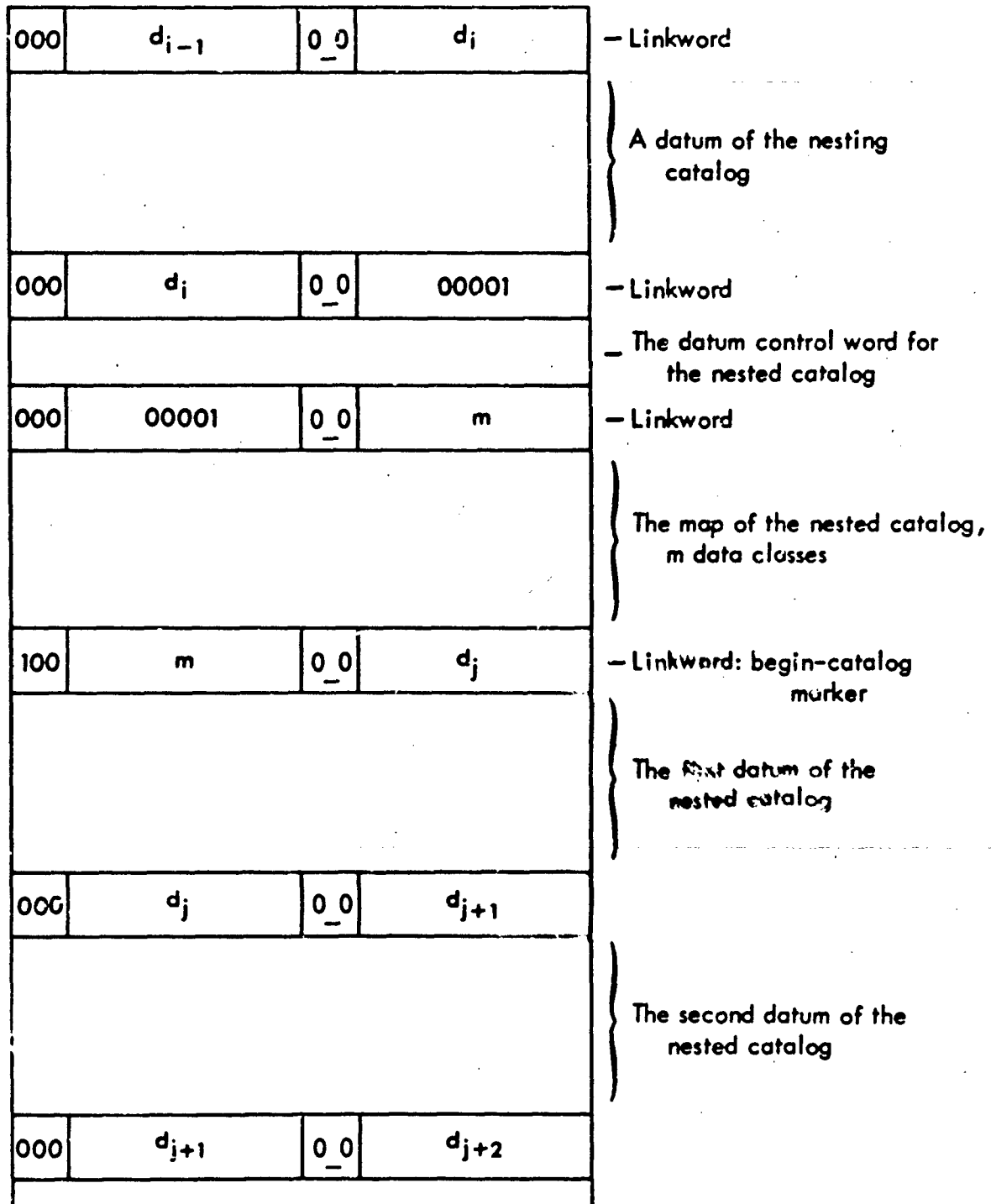


Fig.6 — Beginning of a nested catalog

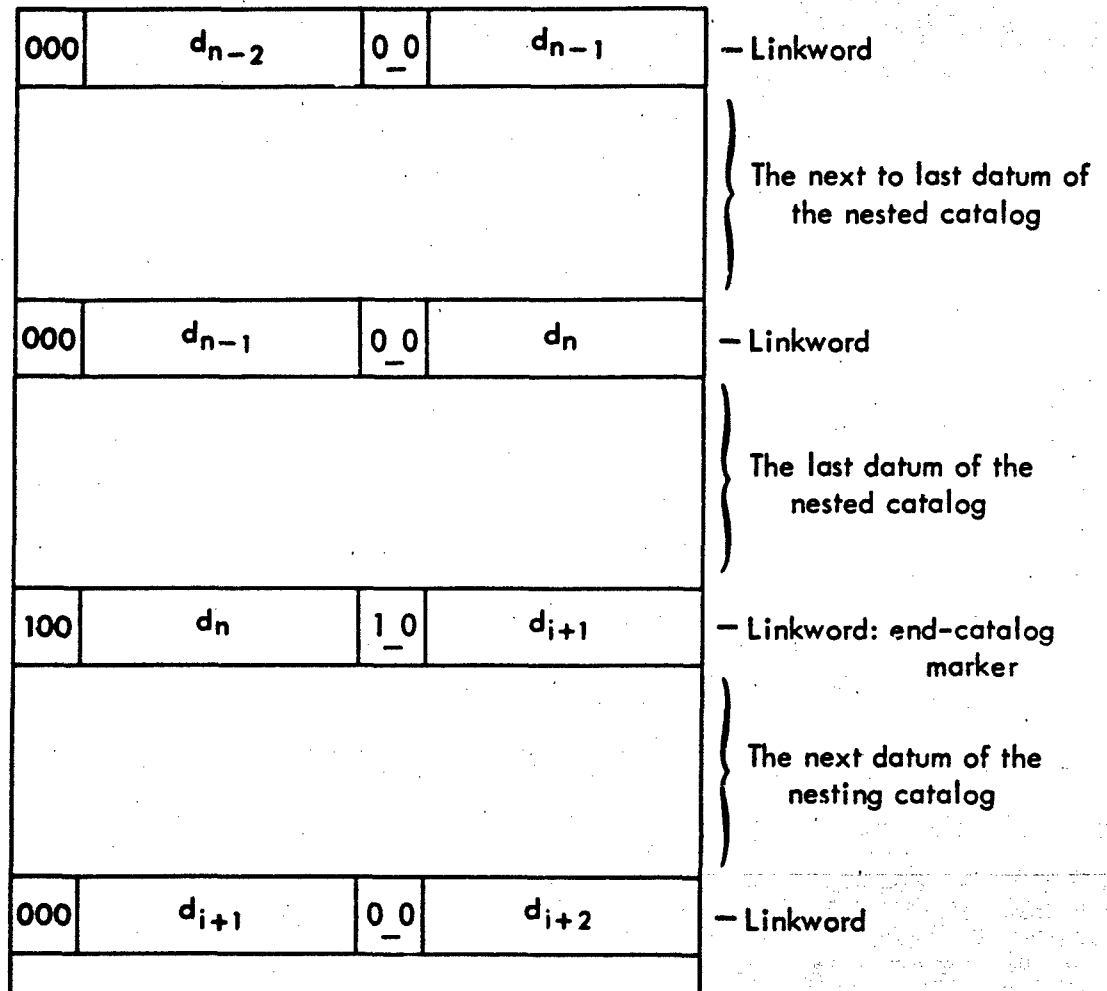


Fig.7 — End of a nested catalog

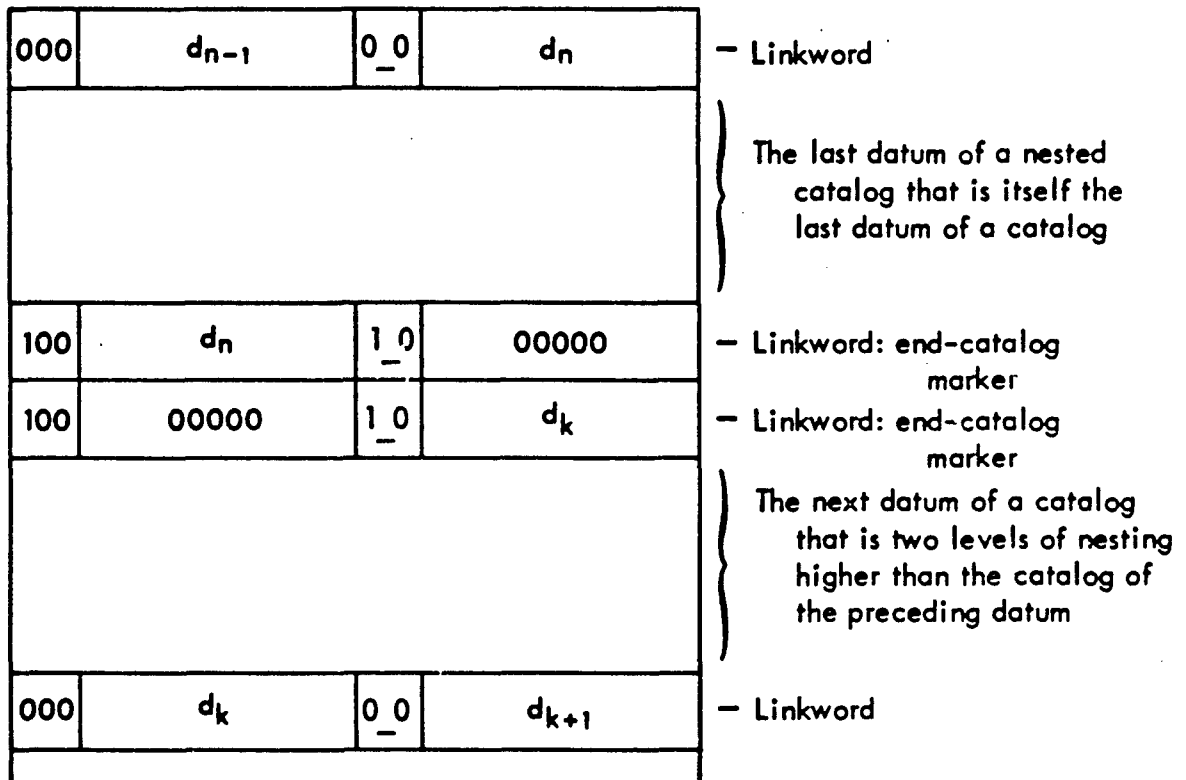


Fig.8 — End of a nested catalog that is itself the last datum of a catalog

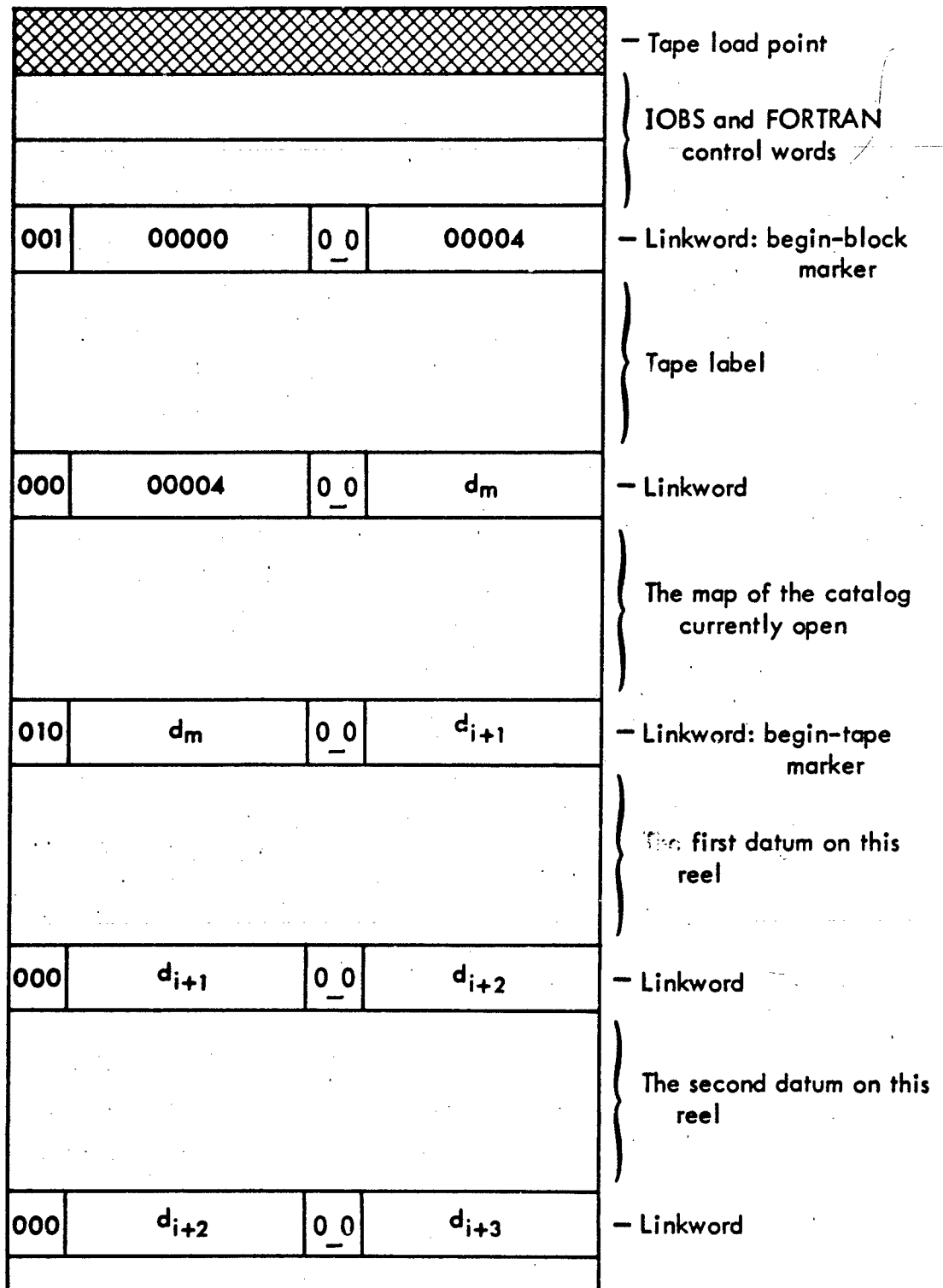


Fig.9 — Beginning of a continuation reel of tape

# MENT CONTROL DATA

ATING ACTIVITY

THE RAND CORPORATION

2a. REPORT SECURITY CLASSIFICATION  
UNCLASSIFIED

2b. GROUP

PORT TITLE

THE CATALOG INPUT/OUTPUT SYSTEM

THOR(S) (Last name, first name, initial)

Kay, Martin, Frederick Valadez and Theodore Ziehe

PORT DATE

March 1966

6a. TOTAL NO. OF PAGES

71

6b. NO. OF REFS.

- - -

TRACT or GRANT NO.

AF 49(638)-1700

8. ORIGINATOR'S REPORT NO.

RM-4540-PR

ILABILITY/LIMITATION NOTICES

DDC 1

9b. SPONSORING AGENCY

United States Air Force  
Project RAND

STRACT

A complete definition of the format  
ed for catalogs on magnetic tape and a  
scription of the routines of the Catalog  
ut/Output System. Catalog maps,  
alog data, and tape labels are written  
logical records in a specially designed  
cking format. Beginnings and ends of  
cks, physical tapes, and catalogs are  
licitly marked. The Catalog Input/  
put System offers a variety of input/  
put unit-control operations in  
dition to the commands for reading and  
ting data in catalog format. The  
ading and writing commands are imple-  
ted on three levels: Level III handles  
ividual data; Level II reads and writes  
gical records; and Level I processes  
cks of information.

II. KEY WORDS

Computer programs  
Catalogs  
MAP language  
Computer language  
IBM 7044  
Data processing  
Information Storage and  
Retrieval  
Linguistics  
Machine Languages  
Higher-level language